



Europäisches  
Patentamt

European  
Patent Office

Office européen  
des brevets

REC'D 29 APR 2004

WIPO

PCT

PHDE 020146 GP P

IB/2004/050511

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

03101255.2 ✓

## PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;  
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets  
p.o.

R C van Dijk



Europäisches  
Patentamt

European  
Patent Office

Office européen  
des brevets

Anmeldung Nr:  
Application no.: 03101255.2 ✓  
Demande no:

Anmeldetag:  
Date of filing: 06.05.03 ✓  
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

Philips Intellectual Property & Standards  
GmbH  
Steindamm 94  
20099 Hamburg  
ALLEMAGNE  
Koninklijke Philips Electronics N.V.  
Groenewoudseweg 1  
5621 BA Eindhoven  
PAYS-BAS

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:  
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.  
If no title is shown please refer to the description.  
Si aucun titre n'est indiqué se référer à la description.)

Verfahren zum zeitlichen Synchronisieren von Uhren

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s)  
revendiquée(s)  
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/  
Classification internationale des brevets:

H04L7/00

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of  
filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL  
PT RO SE SI SK TR LI

BESCHREIBUNG

## Verfahren zum zeitlichen Synchronisieren von Uhren

Die Erfindung betrifft ein Verfahren zum zeitlichen Synchronisieren von Uhren, die über ein Kommunikationsmedium kommunizierenden Knoten zugeordnet sind.

5

Die Erfindung betrifft ferner einen Knoten, der mittels eines Kommunikationsmediums mit anderen Knoten kommuniziert und ein Kommunikationssystem, das mehrere über ein Kommunikationsmedium kommunizierende Knoten aufweist.

- 10 Die Erfindung betrifft auch ein Computerprogramm, das auf einem Recheng Gerät, insbesondere auf einem Mikroprozessor, ablauffähig ist.

Knoten, die über ein Kommunikationsmedium kommunizieren, sind beispielsweise Steuergeräte, Sensoren und Aktoren, die mittels eines Bussystems oder eines Netzwerks miteinander

15 Nachrichten austauschen. Ein aus derartigen Knoten und dem zugehörigen Kommunikationsmedium bestehendes Kommunikationssystem kann beispielsweise zur Steuerung eines Flugzeuges oder eines Kraftfahrzeuges eingesetzt werden. Dabei umfasst der Begriff „Knoten“ insbesondere auch sogenannte Interfaces, die sowohl für eine physikalische Anbindung an das Kommunikationsmedium als auch eine Aufbereitung der gesendeten und empfangenen

- 20 Nachrichten verantwortlich sein können.

Ein modernes Kommunikationssystem ist meist fehlertolerant ausgelegt, so dass auch bei Auftreten von Fehlern innerhalb des Kommunikationssystems ein korrektes Ausführen zumindest bestimmter Grundfunktionen gewährleistet ist. Dies ist beispielsweise dann besonders

25 wichtig, wenn das Kommunikationssystem in sicherheitskritischer Umgebung, wie es beispielsweise die Steuerung eines Flugzeuges oder die Steuerung eines Kraftfahrzeuges ist, eingesetzt

wird. Solche Grundfunktionen stellen dann beispielsweise sicher, dass ein Flugzeug auch bei einem Auftreten von Fehlern innerhalb des Kommunikationssystems noch steuerbar bleibt.

- 5 Beispielsweise ist ein fehlertolerantes Kommunikationssystem derart ausgestaltet, dass bei der Behandlung eines innerhalb eines Knotens oder bei der Übertragung von Nachrichten aufgetretenen Fehlers sichergestellt werden, dass dieser Fehler keine Folgefehler beispielsweise dadurch nach sich zieht, dass weitere Knoten aufgrund einer falsch gesendeten, falsch übertragenen oder falsch empfangenen Nachricht nun ihrerseits falsche Nachrichten produzieren
- 10 oder an einer weiteren, regulären Ausführung ihrer Funktionen gehindert werden. Ohne einen solchen Mechanismus kann es beispielsweise geschehen, dass ein Knoten (sogenannter „babbling idiot“) fälschlicherweise eine so große Anzahl von Nachrichten über das Kommunikationsmedium überträgt, dass eine aktive Störung der übertragenen Daten anderer Knoten auftreten kann und/oder das Kommunikationsmedium überlastet wird und somit den übrigen
- 15 Knoten nicht mehr oder zumindest nicht mehr ausreichend zur Verfügung steht.

- In gegenwärtig eingesetzten Kommunikationssystemen wird insbesondere in sicherheitskritischen Umgebungen zunehmend ein sogenanntes zeit-gesteuertes Übertragungsverfahren bei Kommunikationsmedien verwendet, um beispielsweise das Entstehen von Folgefehlern zu
- 20 reduzieren. Dabei ist es typischerweise einem Knoten nur möglich, zu bestimmten Zeiten eine Nachricht zu senden. Damit wird außerdem gewährleistet, dass das Kommunikationsmedium stets genügend Kapazitäten hat, um beispielsweise für sicherheitsrelevante Anwendungen besonders wichtige Nachrichten übertragen zu können.

- 25 Viele Kommunikationssysteme benötigen für eine korrekte Ausführung eine sogenannte globale Zeit. Dies ist beispielsweise wichtig, um die Gleichzeitigkeit von bestimmten Ereignissen, wie eine simultane Änderung von erfassten Werten verschiedener Sensoren, feststellen zu können. Eine globale Zeit ist insbesondere auch für die zeit-gesteuerte Nachrichtenübertragung

notwendig, beispielsweise um festzustellen, wann ein Knoten senden darf. Außerdem ist es bei manchen zeit-gesteuerten Kommunikationsmedien möglich, anhand des Zeitpunktes einer empfangenen Nachricht auf den Sender dieser Nachricht zu schließen.

5

Für die Bestimmung der globalen Zeit sind verschiedene Verfahren bekannt. Beispielsweise existiert in einem Kommunikationssystem eine zentrale Uhr, deren aktueller Wert mittels des Kommunikationsmediums an die entsprechenden Knoten übertragen wird. Dieses Verfahren hat jedoch den Nachteil, dass eine Fehlfunktion der globalen Uhr, beispielsweise dadurch, dass diese konstant denselben Wert als aktuellen Wert ausgibt, meist eine Fehlfunktion des Kommunikationssystems zur Folge hat. Auch ein Übertragungs- oder Empfangsfehler einer ursprünglich korrekt gesendeten globalen Zeit kann bei zwei Knoten zu einer unterschiedlichen Interpretation der aktuellen Zeit führen und damit eine Fehlfunktion auslösen bzw. eine korrekte Kommunikation verhindern.

15

Üblicherweise werden deshalb mehrere Uhren eingesetzt. Beispielsweise kann jeder Knoten eine eigene Uhr besitzen. Durch Ungenauigkeiten der Uhren und durch Fehler, die durch eine gestörte Uhr verursacht werden können, weichen die von diesen Uhren ermittelten aktuellen Zeiten jedoch normalerweise im Laufe der Zeit mehr und mehr voneinander ab. Zum Ermitteln einer einheitlichen globalen Zeit wird deshalb eine Synchronisierung der Uhren durchgeführt. Dabei werden insbesondere fehlertolerante Verfahren zur Synchronisierung der Uhren eingesetzt, bei denen gewährleistet wird, dass von einer bestimmten Anzahl von Uhren fehlerhaft gesendete, übermittelte oder empfangene aktuelle Zeiten nicht die Synchronisierung einer Uhr negativ beeinflussen kann. Insbesondere sollen solche Verfahren auch ermöglichen, dass eine Uhr, die einen fehlerhaften Wert liefert oder die neu gestartet wurde, wieder mit den anderen Uhren synchronisiert werden kann.

25

Bei Zeit-basierten Verfahren zur Übertragung von Nachrichten auf einem Kommunikationssystem werden die einzelnen Nachrichten in Zeitfenstern, sogenannten frames, übertragen.

Dabei werden die frames in Abhängigkeit von der globalen Zeit über das Kommunikationsmedium derart übertragen, dass jedem frame eine globale Zeit eindeutig zugeordnet werden kann. Damit wird ermöglicht, dass jedes frame von jedem Knoten anhand der globalen Zeit eindeutig identifiziert werden kann. Insbesondere kann jeder Knoten daran erkennen, welches frame ihm für das Senden einer Nachricht zugeordnet ist. Wird die globale Uhr in Abhängigkeit der eigenen Uhr ermittelt, so können die von verschiedenen Knoten zwischen zwei Synchronisierungszyklen ermittelten globalen Zeiten abweichen. Um dennoch eine eindeutige Zuordnung eines frames zu einer globalen Zeit gewährleisten zu können, muss zwischen den einzelnen frames eine Zeitspanne liegen, die abhängig ist von der Größe der möglichen maximalen Abweichung zweier Uhren zwischen zwei Synchronisierungszyklen. Da in dieser Zeitspanne, dem sogenannten inter frame gap, jedoch keine Nachrichten übertragen werden können, beeinflusst die Länge der inter frame gaps die Bandbreite (Menge der übertragbaren Daten pro Zeiteinheit) des Kommunikationsmediums. Daraus folgt wiederum, dass die Bandbreite von der Genauigkeit der Uhren abhängig ist.

Die mittlere Genauigkeit der Uhren ist abhängig davon, wie groß ein Zeitintervall zwischen zwei aufeinanderfolgenden Synchronisierungen ist, also „wie lange“ und damit auch „wie weit“ die aktuellen Zeiten der Uhren voneinander abweichen, bis eine erneute Synchronisierung erfolgt. Die mittlere Genauigkeit der Uhren ist auch abhängig davon, wie schnell eine Synchronisation durchgeführt werden kann, da zwischen einer gesendeten Uhrzeit und dem eigentlichen Korrigieren einer Uhr mittels eines während des Synchronisierungsverfahrens ermittelten Korrekturwertes die Uhren bereits wieder auseinandergedriftet sein können.

Zur Synchronisierung von Uhren existieren Verfahren, die auf der Bildung eines Durchschnitts (sogenannten durchschnitt-basierte Verfahren) beruhen. Bei durchschnitt-basierten Verfahren, wie sie beispielsweise bei den Bussystemen TTA und FlexRay eingesetzt werden, ermittelt jeder zu synchronisierende Knoten den Durchschnitt von aktuell erfassten Zeiten bzw.

Zustandswerten, die diesem von verschiedenen anderen Uhren übermittelt werden. Um bei diesen Verfahren eine Unabhängigkeit von einer bestimmten Anzahl von falschen Uhrzeiten, die an einen Knoten übermittelt werden, zu erreichen, wird eine sogenannte Fehlertoleranz  $k$  gewährleistet. Dabei bleiben beispielsweise ein oder mehrere, von den Zustandswerten der übrigen Uhren extrem abweichende Zustandswerte (beispielsweise die  $k$  größten und die  $k$  kleinsten Zustandswerte), bei der Ermittlung des Durchschnitts unberücksichtigt.

Insbesondere die durchschnitt-basierten Verfahren setzen für eine fehlertolerante Synchronisation von Uhren eine bestimmte minimale Anzahl  $n$  von ermittelten aktuellen Zustandswerten, beispielsweise  $n \geq (2k+2)$ , voraus. Werden diese nicht erreicht, so kann entweder keine Synchronisation durchgeführt werden oder es sind zeitaufwendige Verfahren notwendig, um eine Synchronisation zu gewährleisten.

Aufgabe der Erfindung ist es, eine Möglichkeit bereitzustellen, die es erlaubt, besonders schnell und effizient eine möglichst fehlertolerante, zeitliche Synchronisation von Uhren durchzuführen.

Die Aufgabe wird durch ein Verfahren der eingangs genannten Art gelöst, das die nachfolgenden Schritte aufweist:

- zumindest für die zu synchronisierenden Knoten: erfassen von Zustandswerten, welche abhängig sind von einer Zeitbasis der Knoten;
- für alle erfassten Zustandswerte: Einsortieren des erfassten Zustandswertes an eine entsprechende Position in einer ersten,  $(k+1)$  Positionen umfassenden Liste  $L$ , falls der erfasste Zustandswert kleiner als das  $(k+1)$ -kleinste Element oder kleiner gleich dem  $(k+1)$ -kleinsten Element der Liste  $L$  ist und wobei  $k$  eine vorgebbare Fehlertoleranz ist;

- für alle erfassten Zustandswerte: Einsortieren des erfassten Zustandswertes  $Z$  an eine entsprechende Position in einer zweiten,  $(k+1)$  Positionen umfassenden Liste  $H$ , falls der erfasste Zustandswert größer als das  $(k+1)$ -größte Element oder größer gleich dem
- 5  $(k+1)$ -größten Element der Liste  $H$  ist;
- bilden eines Mittelwertes aus dem  $(k+1)$ -kleinsten Element der Liste  $L$  und dem  $(k+1)$ -größten Element der Liste  $H$ , falls  $n \geq (2k+2)$  gilt, wobei  $n$  die Anzahl der erfassten Zustandswerte ist;
- ermitteln eines Korrekturwertes in Abhängigkeit des Mittelwertes; und
- 10 - korrigieren der zu synchronisierenden Uhren dahingehend, dass ein aktueller Zustandswert dieser Uhr den Korrekturwert berücksichtigt.

#### Vorteile der Erfindung

- 15 Bei dem erfindungsgemäßen Verfahren werden die Zustandswerte der Uhren anderer Knoten sowie ein Zustandswert der eigenen Uhr in die erste Liste  $L$  einsortiert, wenn sie kleiner oder kleiner gleich dem größten dort abgespeicherten Zustandswert sind. Außerdem werden die Zustandswerte dann in die zweite Liste  $H$  einsortiert, wenn sie größer als das kleinste oder größer gleich dem kleinsten in der Liste  $H$  bereits abgespeicherten Zustandswert sind. Dabei
- 20 umfassen sowohl die Liste  $L$  als auch die Liste  $H$  genau  $(k+1)$  Positionen, so dass in die Liste  $L$  und in die Liste  $H$  jeweils  $(k+1)$  Zustandswerte abgespeichert werden können.

- Mit diesem Verfahren wird ermöglicht, dass nach Ablauf des Verfahrens die  $(k+1)$  kleinsten Zustandswerte in der Liste  $L$  in Abhängigkeit ihrer Größe einsortiert sind und in der Liste  $H$
- 25 genau die  $(k+1)$  größten Werte der erfassten Zustandswerte einsortiert sind. Dabei können die einzelnen Positionen der Listen  $L$  und  $H$  beispielsweise derart durchnummeriert sein, dass sich der  $(k+1)$ -kleinste Zustandswert in der Liste  $L$  an der Position  $L_0$  befindet und der



kleinste erfasste Zustandswert an der Position  $L_k$  abgespeichert ist. Entsprechend kann der  $(k+1)$ -größte Zustandswert in der Liste  $H$  an der Position  $H_0$  und der größte erfasste Zustandswert an der Position  $H_k$  abgespeichert sein.

5

Zur Ermittlung des Mittelwertes aus dem  $(k+1)$ -kleinsten und dem  $(k+1)$ -größten Zustandswert wird beispielsweise das arithmetische Mittel aus den an den Positionen  $L_0$  und  $H_0$  der Listen  $L$  und  $H$  abgespeicherten Zustandswerte gebildet. In Abhängigkeit dieses Mittelwertes wird dann ein Korrekturwert derart ermittelt, dass die dem Knoten zugeordnete Uhr synchronisiert werden kann. Dazu kann der Korrekturwert beispielsweise ein absoluter Wert sein, der der Uhr zugewiesen wird. Es ist auch denkbar, dass der Korrekturwert ein relativer Wert ist, der zu dem aktuellen Wert der dem Knoten zugeordneten Uhr addiert werden muss, um einen mit den anderen Uhren synchronisierten aktuellen Wert zu erhalten.

- 15 Ein besonderer Vorteil des erfindungsgemäßen Verfahren liegt darin, dass nicht für jeden Zustandswert eine Position in der Liste vorgesehen ist. Es ist vielmehr ausreichend, wenn lediglich in jeder Liste  $(k+1)$  Positionen vorhanden sind, insgesamt also lediglich  $(2k+2)$  Zustandswerte sortiert und gespeichert werden müssen. Alle Zustandswerte, die größer als der  $(k+1)$ -kleinste Zustandswert und kleiner als der  $(k+1)$ -größte Zustandswert sind, können
- 20 folglich verworfen werden. Damit kann eine speicherplatz-sparende Erfassung und Verwaltung der für die Berechnung des Mittelwertes notwendigen Zustandswerte erreicht werden.

Mit dem erfindungsgemäßen Verfahren kann folglich auch die Anzahl der Speicherzugriffe verringert werden. Somit kann ein besonders effizientes und schnelles Verfahren zur Synchronisierung von Uhren insbesondere dann erreicht werden, wenn die Zahl der erfassten Zustandswerte deutlich größer als  $(2k+2)$  ist.

25

In modernen Bussystemen, wie beispielsweise dem FlexRay, kann beispielsweise die Fehler-toleranz  $k=2$  betragen. In diesem Fall müssen bei dem erfindungsgemäßen Verfahren lediglich

- die Liste L und die Liste H jeweils drei Positionen umfassen, nämlich  $L=\{L0, L1, L2\}$  und  $H=\{H0, H1, H2\}$ . Damit werden maximal sechs Zustandswerte sortiert abgespeichert, selbst wenn die Anzahl der insgesamt von einem zu synchronisierten Knoten erfassten und für die
- 5 Ermittlung des Korrekturwertes zu berücksichtigenden Zustandswerte deutlich über sechs liegt, was beispielsweise in Implementierungen des FlexRay-Bussystems zur Steuerung eines Kraftfahrzeuges häufig der Fall ist.

- In einer vorteilhaften Weiterbildung des Verfahrens wird das Einsortieren der ermittelten
- 10 Zustandswerte in der ersten Liste L und/oder in der zweiten Liste H sequentiell durchgeführt. Da die Zustandswerte von einem Knoten in der Regel sequentiell erfasst werden, ermöglicht es diese Weiterbildung, einen erfassten Zustandswert direkt dahingehend weiterzubearbeiten, dass überprüft wird, ob dieser erfasste Zustandswert in eine der Listen L und/oder H einsortiert werden soll, oder ob er verworfen werden soll. Es muss folglich keine Zwischenspeicherung mehrerer Zustandswerte vorgenommen werden, wodurch erneut Speicherplatz eingespart wird. Durch die damit verbundene verminderte Anzahl von Speicherzugriffen kann
- 15 nochmals eine Beschleunigung des Verfahrens erreicht werden.

- Ein weiterer Vorteil dieser Weiterbildung ist, dass die Zeitspanne zwischen dem Erfassen des
- 20 ersten Zustandswertes und dem Abschluss der Synchronisation verringert werden kann, da jeder Zustandswert sequentiell bereits verarbeitet ist, wenn ein folgender Zustandswert erfasst wird. Damit reduziert sich das Verfahren nach Erfassen des letzten Zustandswertes auf das Einsortieren des letzten Zustandswertes in die Liste L und/oder die Liste H, das Bilden des Mittelwertes aus den an den Positionen L0 und H0 abgespeicherten Zustandswerten, das
- 25 Ermitteln des Korrekturwertes und das Korrigieren der Uhr. Insbesondere sind bei dieser Weiterbildung bei Erfassung des letzten Zustandswertes in der Regel bereits eine Vielzahl der erfassten Zustandswerte verworfen.

In einer bevorzugten Ausführungsform wird die Liste L durch entsprechende Register L0, L1, ..., Lk und/oder die Liste H durch entsprechende Register H0, H1, ..., Hk realisiert. Damit kann eine besonders schnelle Synchronisation der Uhren erreicht werden, da ein Register  
 5 typischerweise in Hardware realisierbar ist und somit einen besonders schnellen Schreib- und Lesezugriff erlaubt.

Vorteilhafterweise wird die erste Liste L mit Werten initialisiert, die größer als der größte zu erwartende Zustandswert sind und die zweite Liste H mit Werten initialisiert, die kleiner als  
 10 der kleinste zu erwartende Zustandswert sind. Mit einer derartigen Initialisierung der Listen L und H wird erreicht, dass für alle Zustandswerte eine Einsortierung in die Liste L und/oder in die Liste H von denselben Bedingungen abhängig ist. Diese Bedingungen sind in Anlehnung an die oben bereits genannten Bedingungen:

- Einsortieren eines Zustandswertes in die erste Liste L, falls der Zustandswert kleiner oder  
 15 kleiner gleich dem größten sich in der Liste L befindlichen Wert ist.
- Einsortieren eines Zustandswertes in die zweite Liste H, falls der Zustandswert größer oder größer gleich dem kleinsten sich in der Liste L befindlichen Wert ist.

Mit diese Ausführungsform können diese Bedingungen unabhängig davon formuliert werden,  
 20 ob bereits Zustandswerte in den Listen L und H einsortiert sind.

Vorzugsweise bleibt bei dem Einsortieren eines erfassten Zustandswertes in die erste Liste L eine Sortierung bezüglich der Größe der abgespeicherten Zustandswerte erhalten, so dass stets  $\text{Wert}(L_0) \geq \text{Wert}(L_1) \geq \dots \geq \text{Wert}(L_k)$  gilt, wobei L0, l1, ..., Lk die (k+1) Positionen  
 25 der Liste L bezeichnen und  $\text{Wert}(L_i)$  der Wert an der Position Li ist. Ferner bleibt bei dem Einsortieren eines erfassten Zustandswertes in die zweite Liste H eine Sortierung bezüglich der Größe der abgespeicherten Zustandswerte erhalten, so dass stets  $\text{Wert}(H_0) \leq \text{Wert}(H_1) \leq \dots \leq \text{Wert}(H_k)$  gilt, wobei H0, H1, ..., Hk die (k+1) Positionen der Liste L bezeichnen und  $\text{Wert}(H_i)$  der Wert an der Position Hi ist.

Damit wird besonders einfach erreicht, dass eine Sortierung der Listen L und H stets gewährleistet ist und nicht beispielsweise erst nach Erfassen des letzten zu berücksichtigenden Zustandswertes durch den Knoten ein zeitaufwändiges Sortieren durchgeführt werden muss.

- 5 Insbesondere in Verbindung mit der oben beschriebenen sequentiellen Bearbeitung der erfassten Zustandswerte wird damit auch sichergestellt, dass der (k+1)-kleinste der erfassten Zustandswerte stets an der Position L0 und der (k+1)-größte der erfassten Zustandswerte stets an der Position H0 abgespeichert ist.
- 10 Weitere Vorteile ergeben sich aus den in den Ansprüchen 6 und 7 beschriebenen Ausführungsformen. Dort wird beschrieben, wie ein Einsortieren eines Zustandswertes Z in die erste Liste L und/oder in die zweite Liste H besonders einfach und übersichtlich realisiert werden kann. Dazu wird zunächst eine der Größe des Zustandswertes Z entsprechende Position innerhalb der Listen L und H durch Vergleich mit den in den Listen befindlichen
- 15 Elementen ermittelt, beispielsweise beginnend mit dem größten bzw. kleinsten bisher einsortierten Element. Der Zustandswert Z wird dann dadurch einsortiert, dass alle größeren Elemente der Liste L eine Position nach „unten“ verschoben werden, wobei der größte in der Liste L sich an der Position L0 befindliche Wert aus der Liste L dadurch herausfällt, dass der Wert an der Position L0 durch den sich an der Position L1 befindlichen Wert ersetzt wird.
- 20 Gleiches gilt analog für das Einsortieren eines Zustandswertes Z in die Liste H.

In einer bevorzugten Ausführungsform werden die folgenden Schritte durchgeführt:

- in Abhängigkeit der Fehlertoleranz k wird eine Menge B von vorgebbaren Endwerten  $B = \{B_0, B_1, \dots, B_k\}$  derart vorgegeben, dass  $B_0 = 0$ ;  $B_i \leq B_{i+1}$ , für alle  $i \in \{0, 1, \dots, (k-1)\}$ ; und  $2j < B(j)$ , für alle  $j \in \{1, \dots, k\}$  gilt;
- 25 - falls  $B_k > n$  gilt, wird in Abhängigkeit der Anzahl n der ermittelten Zustandswerte ein Wert i für  $i \in \{0, 1, \dots, (k-1)\}$  gewählt, so dass die Bedingung  $B_i \leq n < B_{i+1}$  gilt;

- falls  $B_k \leq n$  gilt, wird  $i = k$  gewählt; und
- der Mittelwert wird aus den an den Positionen  $L(k-i)$  und  $H(k-i)$  abgespeicherten Werten gebildet.

5

- Diese Ausführungsform ist geeignet, auch bei Unterschreiten der in dem Knoten erfassten Anzahl  $n$  von Zustandswerten derart, dass diese Anzahl  $n$  kleiner als  $(2k+2)$  ist, eine Synchronisierung der Uhren zu gewährleisten. Insbesondere wird dabei eine tatsächliche Fehlertoleranz  $l$  erreicht, die nicht fest vorgegeben ist, sondern in Abhängigkeit der Anzahl  $n$  der erfassten Zustandswerte, in Abhängigkeit der vorgebbaren Fehlertoleranz  $k$  und in Abhängigkeit der vorgebbaren Endwerte  $B_0, B_1, \dots, B_k$  bestimmt wird, wobei stets  $l \leq k$  gilt. Damit hat diese Ausführungsform den Vorteil, auch bei weniger als der ursprünglich erwarteten Anzahl von erfassten Zustandswerten in einem Knoten, eine Synchronisation der Uhr in diesem Knoten zu gewährleisten. Insbesondere ist bei dieser Ausführungsform im Falle  $n <$
- 10 (2k+2) eine fehlertolerante Synchronisation mit einer Fehlertoleranz  $l \leq k$  möglich.
- 15

Vorteilhafterweise werden die folgenden Werte vorgegeben:

- Fehlertoleranz  $k = 2$ ;
- Endwert  $B_1 = 3$ ; und
- 20 - Endwert  $B_2 = 8$ .

Mit diesen Werten kann besonders gut und schnell eine fehlertolerante Synchronisation von Uhren in einem Kommunikationssystem, beispielsweise FlexRay, für jede Anzahl  $n$  von erfassten Zustandswerten gewährleistet werden.

25 Die Aufgabe wird auch durch einen Knoten gelöst, der

- eine Uhr aufweist;
- Mittel zum Erfassen von Zustandswerten aufweist, wobei die Zustandswerte von einer Zeitbasis des Knotens und/oder von einer Zeitbasis der anderen Knoten abhängen;

- eine  $(k+1)$  Positionen umfassende erste Liste (L) und eine  $(k+1)$  Positionen umfassende zweite Liste (H) aufweist;
- Mittel zum Einsortieren eines erfassten Zustandswertes an eine entsprechende Position der ersten Liste (L) aufweist;
- 5 - Mittel zum Einsortieren eines erfassten Zustandswertes an eine entsprechende Position der zweiten Liste (H) aufweist;
- Mittel zum Bilden eines Mittelwertes aus einem Element der ersten Liste (L) und einem Element der zweiten Liste (H) aufweist;
- Mittel zum Bilden eines Korrekturwertes aufweist; und
- 10 - Mittel zum Korrigieren der Uhr aufweist.

Vorzugsweise wird dabei in dem Knoten ein Verfahren nach einem der Ansprüche 1 bis 9 ausgeführt.

- 15 Die Aufgabe wird ferner durch ein Kommunikationssystem der eingangs genannten Art gelöst, bei dem mindestens ein Knoten
  - eine Uhr (15) aufweist;
  - Mittel zum Erfassen von Zustandswerten aufweist;
  - eine  $(k+1)$  Positionen umfassende erste Liste (L) und eine  $(k+1)$  Positionen umfassende
  - 20 zweite Liste (H) aufweist;
  - Mittel zum Einsortieren (120) eines erfassten Zustandswertes an eine entsprechende Position der ersten Liste (L) aufweist;
  - Mittel zum Einsortieren (130) eines erfassten Zustandswertes an eine entsprechende Position der zweiten Liste (H) aufweist;
  - 25 - Mittel zum Bilden (160) eines Mittelwertes (M) aus einem Element der ersten Liste (L) und einem Element der zweiten Liste (H) aufweist;
  - Mittel zum Bilden eines Korrekturwertes (K) aufweist; und
  - Mittel zum Korrigieren der Uhr (15) aufweist.

Vorzugsweise wird bei dem Kommunikationssystem in mindestens einem Knoten ein Verfahren nach einem der Ansprüche 1 bis 9 ausgeführt.

Von besonderer Bedeutung ist die Realisierung dieser Erfindung in Form eines Computer-  
5 programms. Dabei ist das Computerprogramm auf einem Recheng Gerät, insbesondere auf einem Mikroprozessor ablauffähig und zur Ausführung des erfindungsgemäßen Verfahrens geeignet. In diesem Fall wird also die Erfindung durch das Computerprogramm realisiert, so dass dieses Computerprogramm in gleicher Weise die Erfindung darstellt wie das Verfahren, zu dessen Ausführung das Computerprogramm geeignet ist. Das Computerprogramm ist  
10 vorzugsweise auf einem Speicherelement abgespeichert. Als Speicherelement kann insbesondere ein Random-Access-Memory, ein Read-Only-Memory oder ein Flash-Memory zur Anwendung kommen.

Ebenfalls von Bedeutung ist die Realisierung dieser Erfindung in Hardware, so dass die Hard-  
15 ware zur Ausführung des erfindungsgemäßen Verfahrens geeignet ist. Eine Realisierung in Hardware hat den Vorteil einer noch schnelleren Ausführung des erfindungsgemäßen Verfahrens.

Weitere Merkmale, Anwendungsmöglichkeiten und Vorteile der Erfindung ergeben sich aus  
20 der nachfolgenden Beschreibung von Ausführungsbeispielen der Erfindung, die in der Zeichnung dargestellt sind. Dabei bilden alle beschriebenen oder dargestellten Merkmale für sich oder in beliebiger Kombination den Gegenstand der Erfindung, unabhängig von ihrer Zusammenfassung in den Patentansprüchen oder deren Rückbeziehung sowie unabhängig von ihrer Formulierung bzw. Darstellung in der Beschreibung bzw. in der Zeichnung. Es zeigen:

25

- Figur 1            ein Kommunikationssystem bestehend aus mehreren Knoten, die über ein Kommunikationsmedium kommunizieren;
- Figur 2a           eine schematische Darstellung von ausgewählten Komponenten eines Knotens;

- Figur 2b eine schematische Darstellung einer anderen Anordnung der in Figur 2a dargestellten Komponenten eines Knotens;
- Figur 5 eine schematische Darstellung eines Kommunikationszyklus mit mehreren frames;
- Figur 4 ein Ablaufdiagramm eines erfindungsgemäßen Verfahrens;
- Figur 5 eine schematische Darstellung von in Listen L und H abgespeicherten Zustandswerten Z während einer Ausführung des erfindungsgemäßen Verfahrens, wobei  $n < (2k+2)$  ist; und
- 10 Figur 6 eine schematische Darstellung analog zu Figur 5, jedoch mit  $n \geq (2k+2)$ .

In Figur 1 ist ein Kommunikationssystem 1 dargestellt, das aus mehreren Knoten 10 besteht, die über ein Kommunikationsmedium 5 miteinander verbunden sind. Das Kommunikationsmedium 5 ist als zeit-gesteuertes Bussystem (TTA – Time Triggered Architecture), beispielsweise gemäß der FlexRay-Spezifikation oder gemäß dem TTP (Time Triggered Protocol), ausgebildet. Es ist auch denkbar, dass das Kommunikationssystem 5 gemäß einer anderen Netzwerk-Topologie, beispielsweise als ein Ring aufgebaut ist.

Jeder in Figur 1 dargestellte Knoten 10 weist einen sogenannten Host 11 und ein Interface 12 auf. Der Host 11 kann beispielsweise als ein Kleinstrechner ausgestaltet sein, der auf einem einzigen Mikrochip realisiert ist. Es ist ebenso denkbar, dass der Host 11 ein gesamtes Computersystem ist, das seinerseits aus mehreren Computern besteht, die über ein Netzwerk miteinander kommunizieren.

Das Interface 12 befindet sich zwischen dem Host 11 und dem Kommunikationssystem 5. Es regelt beispielsweise die Übertragung von Informationen von dem Host 11 oder an den Host 11 dadurch, dass es die Informationen in ein Nachrichtenformat umwandelt, das den durch das Kommunikationssystem 5 vorgegebenen Datenrahmen (sogenannten frames) entspricht.



Ein frame ist typischerweise eine Menge von bits, die nach vorgegebenen Regeln interpretiert werden, wobei die Regeln durch das Nachrichtenformat definiert sind. Das Nachrichtenformat von FlexRay beispielsweise bezeichnet ein spezielles SYNC-bit, das angibt, ob die betreffende Nachricht zum Synchronisieren verwendet werden soll, und ein LEN-bit, das die Anzahl  
5 der Datenbytes angibt, die die eigentliche Nachricht (Nutzdaten) enthalten.

Selbstverständlich ist es ebenso möglich, dass der Host 11 und das Interface 12 lediglich als unterschiedliche Funktionseinheiten auf demselben Mikrochip ausgebildet sind.

10 In einem zeit-gesteuerten Kommunikationsmedium 5 ist die Bandbreite der Kommunikation in Form eines (oder mehrerer) globaler zeitlicher Abläufe (sogenannten globalen schedules) fest vorgebar. Ein solches schedule umfasst Informationen darüber, wann ein Knoten 10 eine Nachricht mittels des Kommunikationsmediums 5 übertragen kann und wann er eine Nachricht erwarten kann. Werden die Nachrichten beispielsweise in frames übertragen, gibt ein  
15 schedule insbesondere darüber Auskunft, welches frame einem Knoten 10 zum Senden einer Nachricht zur Verfügung steht.

Da die gesamte Kommunikation auf dem Kommunikationsmedium 5 zeit-gesteuert mittels des globalen schedules ist, muss eine Nachricht keine Empfänger- oder Senderadresse enthalten.  
20 Jeder Knoten 10 identifiziert den Sender und Empfänger einer Nachricht daran, wann diese Nachricht gesendet wurde. Damit kann auf ein Adressfeld in den frames verzichtet werden, was die Bandbreite erhöht.

In Figur 2a ist ein Interface 12 dargestellt, das eine Uhr 15 einen Mikroprozessor 20 und ein  
25 Speicherelement 30 aufweist. Das Speicherelement 30 ist beispielsweise als random-access-memory (RAM) ausgebildet und weist adressierbare Speicherbereiche L0, L1, ..., Lk, und H0, H1, ..., Hk auf. Das Interface 12 weist auch ein weiteres Speicherelement 31 auf, das beispielsweise als read-only-memory (ROM) ausgestaltet ist. Auf dem Speicherelement 31

kann beispielsweise ein Computerprogramm abgespeichert sein, das zur Ausführung des erfindungsgemäßen Verfahrens programmiert ist. Selbstverständlich ist es auch denkbar, dass die Speicherelemente 30 und 31 als ein einziges Speicherelement realisiert sind.

5

Ein in Figur 2b dargestelltes Interface 12 weist ebenfalls eine Uhr 15, einen Mikroprozessor 20 und ein Speicherelement 31 auf. Der Mikroprozessor 20 weist hier jedoch Register L0, L1, ..., Lk und H0, H1, ..., Hk auf. Die hier dargestellte Ausführungsform eines Interfaces 12 hat gegenüber der in Figur 2a dargestellten Ausführungsform den Vorteil, dass die Zugriffszeit eines Registers L0, L1, ..., Lk und H0, H1, ..., Hk kürzer ist als bei anderen Speicherelementen. Dadurch können bei einer Verwendung von Registern L0, L1, ..., Lk und H0, H1, ..., Hk anstelle der in Figur 2a dargestellten entsprechenden Speicherbereiche Daten schneller an die Positionen L0, L1, ..., Lk, H0, H1, ..., Hk geschrieben und von diesen Positionen gelesen werden.

15

In Figur 3 ist ein Kommunikationszyklus 51 dargestellt, dessen Beginn mittels einer gestrichelten Linie 52 und dessen Ende mittels einer gestrichelten Linie 53 gekennzeichnet ist. Zu Beginn eines Kommunikationszyklus 51 wird beispielsweise ein sogenanntes SYNC-Symbol 55 übermittelt, dass es allen an der Kommunikation beteiligten Knoten 10 ermöglicht, den Beginn eines Kommunikationszyklus 51 zu erkennen. Es ist jedoch auch möglich, einen Kommunikationszyklus ohne das SYNC-Symbol 55 zu beginnen. Innerhalb des Kommunikationszyklus 51 werden typischerweise mehrere frames 71, 72, 73, 74, ... übertragen, von denen hier beispielhaft vier frames dargestellt sind. Die frames 71, 72, 73, 74 werden bei einem zeitgesteuerten Übertragungsverfahren innerhalb bestimmbarer sogenannter Zeitslots übertragen, deren Beginn hier mit den Bezugszeichen 61, 62, 63, 64 bezeichnet sind. Zwischen den frames 71, 72, 73, 74 befinden sich Zeitspannen, die sogenannten inter frame gaps 81, 82, 83, 84, in denen keine Nachrichten übertragen werden. Die inter frame gaps 81, 82, 83, 84

25

sind notwendig, um auch bei leicht voneinander abweichenden Uhren 15 der Knoten 10 innerhalb eines Kommunikationssystems 1 eine eindeutige Identifizierung eines frames 71, 72, 73, 74 zu einem dieses frame sendenden Knoten zu ermöglichen.

5

Damit alle über das Kommunikationsmedium 5 kommunizierenden Knoten 10 die frames 71, 72, 73, 74 bezüglich deren Zuordnung zu einem Knoten 10 identisch beurteilen, ist es besonders wichtig, dass die Uhren dieser Knoten 10 synchronisiert sind. Dabei kann die Uhr 15 in dem Host 11 oder in dem Interface 12 lokalisiert sein. Ebenso kann der Prozess der Synchronisierung der Uhr 15 prinzipiell in dem Host 11 oder dem Interface 12 durchgeführt werden.

10

Jedoch hängt die Bandbreite der Kommunikation von der Genauigkeit der Synchronisierung der Uhren 15 in dem Kommunikationssystem 1 ab, da – um eine eindeutige Identifizierung der frames 71, 72, 73, 74, durch alle Knoten 10 zu gewährleisten – die frames 71, 72, 73, 74, nicht unmittelbar aufeinanderfolgen, sondern sich zwischen den frames 71, 72, 73, 74, ein inter frame gap 81, 82, 83, 84 befindet. Dieses inter frame gap 81, 82, 83, 84 bezeichnet eine Zeitspanne, die nach dem Ende eines ersten frames 71, 72, 73, 74, und dem Beginn eines folgenden zweiten frames 71, 72, 73, 74, vergeht. Damit kann bei hinreichend genau synchronisierten Uhren verhindert werden, dass zwei Knoten 10 dasselbe frame 71, 72, 73, 74, aufgrund ihrer aktuellen Zeit hinsichtlich des schedules beispielsweise derart unterschiedlich interpretieren, dass ein Knoten 10, dessen Uhr 15 bezüglich der globalen Uhr nachgeht, das zweite frame 71, 72, 73, 74, fälschlicherweise als das erste frame interpretiert.

15

20

Je genauer die Uhren 15 der Knoten 10 untereinander synchronisiert sind, desto kleiner kann das inter frame gap 81, 82, 83, 84 zwischen den einzelnen frames 71, 72, 73, 74, sein. Je kleiner das inter frame gap 81, 82, 83, 84 zwischen den einzelnen frames 71, 72, 73, 74, ist, desto größer wiederum ist die Bandbreite der Kommunikation.

25

Typischerweise wird die Synchronisierung mittels spezialisierter Hardware durchgeführt, die in das Interface 12 integriert ist, um eine schnelle und exakte Synchronisierung der ebenfalls in die Interfaces 12 integrierten Uhren 15 erreichen zu können.

5

In Figur 4 ist ein sehr vereinfachtes Ablaufdiagramm eines erfindungsgemäßen Verfahrens dargestellt. Das Verfahren startet in einem Schritt 100 beispielsweise dann, wenn in einem Knoten 10 eine Synchronisierung der diesem Knoten 10 zugeordneten Uhr 15 durchgeführt werden soll. Bei dem Start des Verfahrens wird eine Initialisierung bestimmter Variablen durchgeführt. Beispielsweise wird die Anzahl  $n$  der erfassten Zustandswerte auf Null gesetzt ( $n = 0$ ).

In einem Schritt 110 wird ein Zustandswert  $Z$  beispielsweise dadurch erfasst, dass ein mittels des Kommunikationsmediums 5 übermitteltes frame 71, 72, 73, 74, von dem Knoten 10 als Synchronisationsnachricht interpretiert wird und der Zustandswert  $Z$  aus den entsprechenden, in dem Nachrichtenformat definierten, bits des frames 71, 72, 73, 74, ausgelesen wird. Nach Erfassen des Zustandswertes  $Z$  wird die Anzahl  $n$  der erfassten Zustandswerte um 1 erhöht ( $n++$ ).

In einem Schritt 120 wird der erfasste Zustandswert  $Z$  in eine erste Liste  $L$ , die beispielsweise durch die Register  $L0$  bis  $Lk$  realisiert wird, einsortiert, falls der Zustandswert  $Z$  kleiner ist, als der größte, in der Liste  $L$  abgespeicherte Wert. Um zu gewährleisten, dass die ersten  $k$  erfassten Zustandswerte auf jeden Fall in die Liste  $L$  abgespeichert werden, ohne dass eine Überprüfung stattfindet, wie viele Zustandswerte bereits in der Liste  $L$  abgespeichert wurden, werden in der Initialisierungsphase des Schrittes 100 die Register  $L0$  bis  $Lk$  mit einem Wert  $+\infty$  initialisiert, der größer als jeder zu erwartende Zustandswert  $Z$  ist, so dass stets gilt:  $Z < +\infty$ , für alle  $Z$ .

Im folgenden wird ein beispielhaftes, in einem Pseudo-Code notiertes Verfahren für das Einsortieren eines Zustandswertes  $Z$  in die Liste  $L$  dargestellt. Dabei werden die in den Registern  $L_0, \dots, L_k$  abgespeicherten Werte mit  $L[0], \dots, L[k]$  bezeichnet:

```

5
      (11)      for (i = k; i ≥ 0; i--) {
      (12)      if (Z < L[i]) {
      (13)      for (j = 0; j < i; j++) {
      (14)      L[j] = L[j+1];
10      (15)      }
      (16)      L[i] = Z;
      (17)      break i;
      (18)      }
      (19)      }

```

15

In Zeile (11) wird ein Zähler  $i$  initialisiert und eine erste Schleife definiert. Bei jedem erneuten Durchlauf des Schleifenrumpfes wird der Wert  $i$ , beginnend mit  $i = k$ , um den Wert 1 verringert ( $i--$ ). In Abhängigkeit der Wertes von  $i$  werden dann die Instruktionen der Zeilen (12) bis (17) ausgeführt. Durch die erste Schleife in Verbindung mit der Abfrage in Zeile (12),

20 wird die gesamte Liste  $L$  nach einer Position, bzw. einem Register  $L_i$  durchsucht, für das der Zustandswert  $Z$  kleiner ist, als der an der Position bzw. dem Register  $L_i$  abgespeicherte Wert. Wird keine solche Position gefunden, so wird der Wert  $Z$  nicht in die Liste  $L$  einsortiert, sondern verworfen. Wird jedoch in Zeile (12) eine solche Position  $L_i$  erkannt, dann wird in Zeile (13) eine zweite Schleife initialisiert und gestartet, die durch die Zeilen (13) bis (15)

25 begrenzt ist. In dieser zweiten Schleife wird jeder Wert an einer Position  $L_j$  in der Liste durch den Wert  $L_{j+1}$  ersetzt, der an der nächst höheren Position abgespeichert ist, falls  $j < i$  ist. Es werden also alle Werte an Positionen „unterhalb“ der in Zeile (12) gefundenen Position  $L_i$  um

eine Position weiter „nach unten“ geschoben, wobei der Wert an der Position L0 herausfällt. Anschließend wird der Zustandswert  $Z$  in der Zeile (16) an der Position, bzw. in das Register  $L_i$  abgespeichert und das Einsortieren beendet (break i).

5

In einem Schritt 130 des in Figur 4 dargestellten, erfindungsgemäßen Verfahrens wird der erfasste Zustandswert  $Z$  in die Liste  $H$ , die beispielsweise durch die Register  $H_0$  bis  $H_k$  realisiert wird, analog zu Schritt 120 einsortiert, falls der Zustandswert  $Z$  größer ist, als der kleinste, in der Liste  $H$  abgespeicherte Wert. Um zu gewährleisten, dass die ersten  $k$  erfassten

10 Zustandswerte auf jeden Fall in die Liste  $H$  abgespeichert werden, ohne dass eine Überprüfung stattfindet, wie viele Zustandswerte bereits in der Liste  $L$  abgespeichert wurden, werden in der Initialisierungsphase des Schrittes 100 die Register  $L_0$  bis  $L_k$  mit einem Wert  $-\infty$  initialisiert, der kleiner als jeder zu erwartende Zustandswert  $Z$  ist, so dass stets gilt:  $Z > -\infty$ , für alle  $Z$ .

15

Das Einsortieren des Zustandswertes  $Z$  in die Liste  $H$  wird ebenfalls beispielhaft in Pseudo-Code dargestellt. Dabei werden die in den Registern  $H_0$ , ...,  $H_k$  abgespeicherten Werte mit  $H[0]$ , ...,  $H[k]$  bezeichnet:

```

20      (h1)      for (i = k; i ≥ 0; i--) {
                (h2)      if (Z > H[i]) {
                (h3)      for (j = 0; j < i; j++) {
                (h4)      H[j] = H[j+1];
                (h5)      }
25      (h6)      H[i] = Z;
                (h7)      break i;
                (h8)      }
                (h9)      }

```

In Zeile (h1) wird wieder ein Zähler  $i$  initialisiert und eine erste Schleife definiert. In Abhängigkeit der Wertes von  $i$  werden dann die Instruktionen der Zeilen (h2) bis (h7) ausgeführt.

Durch die erste Schleife in Verbindung mit der Abfrage in Zeile (h2), wird die gesamte Liste  $H$  nach einer Position, bzw. einem Register  $H_i$  durchsucht, für das der Zustandswert  $Z$  größer ist,

- 5 als der an der Position bzw. dem Register  $H_i$  abgespeicherte Wert. Wird keine solche Position  $H_i$  gefunden, so wird der Wert  $Z$  nicht in die Liste  $H$  einsortiert, sondern verworfen. Wird jedoch in Zeile (h2) eine solche Position  $H_i$  erkannt, dann wird in Zeile (h3) eine zweite Schleife initialisiert und gestartet, die durch die Zeilen (h3) bis (h5) begrenzt ist. In dieser zweiten Schleife wird jeder Wert an einer Position  $H_j$  in der Liste durch den Wert  $H_{(j+1)}$  ersetzt,
- 10 der an der nächst höheren Position abgespeichert ist, falls  $j < i$  ist. Es werden also alle Werte an Positionen „unterhalb“ der in Zeile (h2) gefundenen Position  $H_i$  um eine Position weiter „nach unten“ geschoben, wobei der Wert an der Position  $H_0$  herausfällt. Anschließend wird der Zustandswert  $Z$  in der Zeile (h6) an der Position, bzw. in das Register  $H_i$  abgespeichert und das Einsortieren beendet (break  $i$ ).

15

In einem Schritt 140 wird überprüft, ob ein neuer Zustandswert  $Z$  erfasst werden soll, beispielsweise, weil ein weiteres SYNC-frame erkannt wurde. Soll ein neuer Zustandswert  $Z$  erfasst werden, so wird das Verfahren in dem Schritt 110 fortgesetzt. Ist dies nicht der Fall, so wird in einem Schritt 150 ein Wert  $j$  in Abhängigkeit der Anzahl  $n$  der erfassten Zustandswerte  $Z$  und in Abhängigkeit von Endwerten  $B_i$  ermittelt. Der Wert  $j$  wird ermittelt, um auch für den Fall, dass die Anzahl  $n$  der erfassten Zustandswerte kleiner als  $(2k+2)$  ist, eine fehler-tolerante Synchronisierung mittels der weiteren Schritte des erfindungsgemäßen Verfahrens zu ermöglichen.

20

25 Der Wert  $j$  wird derart ermittelt, dass stets

$$B_j \leq n < B_{(j+1)}, \text{ für } j \in \{0, 1, \dots, (k-1)\}$$

gilt.

Die Endwerte  $B_i$  (bzw.  $B_j$ ) werden zweckmäßigerweise in der Initialisierungsphase des Schritts 100 initialisiert. Dabei werden die Endwerte  $B_i$  derart initialisiert, dass sie den folgenden Bedingungen genügen:

- 5             $B_0 = 0;$   
               $B_i \leq B_{(i+1)}, \text{ für alle } i \in \{0, \dots, k-1\};$   
               $2j < B_j, \text{ für alle } j \in \{0, \dots, k\};$

Für eine vorgegebene Fehlertoleranz  $k = 2$  könnten die Endwerte  $B_i$  für  $i \in \{0, 1, 2\}$

- 10    beispielsweise wie folgt bestimmt werden:

$$\begin{aligned} B_0 &= 0; \\ B_1 &= 3; \\ B_2 &= 8. \end{aligned}$$

- Sind die Endwerte  $B_j$  wie oben gewählt und werden beispielsweise nur fünf Zustandswerte  $Z$  erfasst, ist folglich die Anzahl  $n = 5$ , so ergibt sich  $j = 1$  für den Wert  $j$ . In Abhängigkeit des Wertes  $j$  wird in einem Schritt 160 ein für die Synchronisierung maßgeblicher Mittelwert  $M$  aus je einem Wert der Liste  $L$  und einem Wert der Liste  $H$  beispielsweise gemäß der .
- Vorschrift

$$M = \frac{L(k-j) + H(k-j)}{2}$$

- 20    (sogenanntes arithmetisches Mittel) bestimmt.

- Für den Fall, dass die Anzahl  $n$  der erfassten Zustandswerte  $n \geq (2k+2)$  gilt, ergibt sich der Wert  $j$  zu  $j = k$ . In diesem Fall wird also der Mittelwert  $M$  aus den an den Positionen  $L_0$  und  $H_0$  abgespeicherten Werten gebildet. Wie oben beschrieben, werden in den Schritten 120 und 130 die Zustandswerte  $Z$  derart in die Listen  $L$  und  $H$  einsortiert, dass sich der kleinste der erfassten Zustandswerte an der Position  $L_k$  und der größte der erfassten Zustandswerte



an der Position  $H_k$  befindet. Für  $n \geq (2k+2)$  werden außerdem der  $(k+1)$ -kleinste Wert an der Position  $L_0$  und der  $(k+1)$ -größte Wert an der Position  $H_0$  abgespeichert. Somit wird im Falle  $n \geq (2k+2)$  Der Mittelwert  $M$  aus dem  $(k+1)$ -kleinsten und dem  $(k+1)$ -größten Wert  
5 der erfassten Zustandswerte bestimmt. Das bedeutet insbesondere, dass die  $k$ -größten und  $k$ -kleinsten erfassten Zustandswerte bei der Bestimmung des Mittelwerts nicht berücksichtigt werden, was im Sinne der Fehlertoleranz  $k$  ist.

Sind die Endwerte  $B_i$  für  $k = 2$  wieder wie oben gezeigt bestimmt und gilt für die Anzahl  $n$  der  
10 erfassten Zustandswerte  $Z$  wieder  $n = 5$ , so ergibt sich der Wert  $j$  wieder zu  $j = 1$  und damit wird in diesem Fall der Mittelwert aus den an den Positionen  $L(k-1) = L(1)$  und  $H(k-1) = L(1)$  abgespeicherten Werten gebildet. Aus diesem Beispiel ist ersichtlich, dass auch im Falle  $n < (2k+2)$  mit dem erfindungsgemäßen Verfahren noch eine fehlertolerante Synchronisation der Uhren möglich ist. In diesem Beispiel beträgt diese variable Fehlertoleranz  $l$  beispielsweise  
15  $l = 1$ .

In Abhängigkeit des Mittelwertes  $M$  wird in einem Schritt 170 ein Korrekturwert  $K$  bestimmt, in einem Schritt 180 wird die dem Knoten 10 zugeordnete Uhr 15 in Abhängigkeit des Korrekturwertes  $K$  synchronisiert und in einem Schritt 190 endet daraufhin das Verfahren.  
20 Der Korrekturwert  $K$  kann beispielsweise ein relativer Wert sein, der den aktuellen Zustandswert der dem Knoten 10 zugeordneten Uhr 15 berücksichtigt, so dass für eine Synchronisierung der Uhr 15 lediglich der Korrekturwert  $K$  zu dem aktuellen Zustandswert addiert werden muss.

25 Selbstverständlich ist es denkbar, dass bei dem in Figur 4 dargestellten Verfahren weitere Schritte hinzukommen oder auf Schritte verzichtet wird. Insbesondere ist es denkbar, dass einzelne Schritte zusammengefasst werden und/oder die Ausführung mehrerer Schritte bzw. die Abarbeitung eines einzelnen Schrittes in einer anderen Reihenfolge geschieht. Ebenso

selbstverständlich ist, dass eine Realisierung des erfindungsgemäßen Verfahrens von den Beispielen, insbesondere den in Pseudo-Code dargestellten Ausführungsbeispielen, abweichen kann.

5

Ein wie in Figur 4 dargestelltes Ausführungsbeispiel eines erfindungsgemäßen Verfahrens kann beispielsweise in Form eines Computerprogramms realisiert werden. Insbesondere um eine beschleunigte Durchführung des erfindungsgemäßen Verfahrens zu erreichen, kann das Verfahren jedoch auch in Hardware, beispielsweise mittels eines sogenannten ASIC

10 (Application-Specific Integrated Circuit), realisiert werden. Es ist ebenso denkbar, eine bereits bestehende Hardware, beispielsweise eine bestimmte Funktionseinheit innerhalb des Interfaces 10, derart zu erweitern, dass eine Durchführung des erfindungsgemäßen Verfahrens möglich ist.

15 Ferner sind sämtliche Bezeichnungen und Indizes, beispielsweise die Nummerierung der Positionen der Listen L und H, selbstverständlich nicht fest, sondern lediglich als logische Bezeichnungen zu sehen. So ist es vorstellbar, dass die Listen L und H sowie Positionen dieser Listen anders bezeichnet werden. Auch eine Änderung der Nummerierung, insbesondere eine Umkehr der Sortierreihenfolge, ist denkbar. Bei einer Umkehrung der Sortier-  
20 reihenfolge wäre beispielsweise der  $(k+1)$ -kleinste Wert an der Position  $L(k)$  und der  $(k+1)$ -größte Wert an der Position  $H(k+1)$  abgespeichert. Entsprechend müssten die Schritte des erfindungsgemäßen Verfahrens angepasst werden, insbesondere die Schritte 120, 130 und 160.

25 In Figur 5 ist ein Beispiel für einen Ablauf des Einsortierens von erfassten Zustandswerten Z in die Listen L und H dargestellt. Dabei geschieht das Einsortieren analog zu der Beschreibung der Schritte 120 und 130 des in Figur 4 dargestellten, erfindungsgemäßen Verfahrens. Dargestellt sind Listen H und L, die Positionen  $H_0$ ,  $H_1$ ,  $H_2$  sowie  $L_0$ ,  $L_1$ ,  $L_2$  aufweisen. Mit Z ist ein erfasster Zustandswert bezeichnet, der in die Listen L und H einsortiert werden soll.

Vor dem Einsortieren eines ersten ( $n = 1$ ) erfassten Zustandswertes  $Z = 5$  sind die Positionen  $L_0, L_1, L_2$  mit dem Wert  $+\infty$  und die Positionen  $H_0, H_1, H_2$  mit dem Wert  $-\infty$  initialisiert. Dabei sind die Werte  $-\infty$  und  $+\infty$  derart gewählt, dass  $-\infty$  kleiner und  $+\infty$  größer ist als jeder zu erfassende Zustandswert  $Z$ , also stets  $-\infty < Z < +\infty$ , für alle  $Z$  gilt.

Zunächst werden, analog zu dem oben in Pseudo-Code beschriebenen und in Figur 4 dargestellten Ausführungsbeispiel Positionen  $L_i$  und  $H_i$  gesucht, an denen der erfasste Wert  $Z$  abgespeichert werden soll. Dabei wird eine solche Position  $L_i$  bzw.  $H_i$  dann erkannt, wenn gilt

10  $L[i] > Z$  bzw.  $H[i] < Z$ , wobei in diesem Ausführungsbeispiel die Listen von oben nach unten, also in der Reihenfolge  $L_2 \rightarrow L_1 \rightarrow L_0$  bzw.  $H_2 \rightarrow H_1 \rightarrow H_0$  durchsucht werden. Für den ersten Zustandswert  $Z = 5$  werden die Positionen  $L_2$  und  $H_2$  ermittelt. Daraufhin werden die Werte an den Positionen  $L_0$  bzw.  $H_0$  durch die Werte an den Positionen  $L_1$  bzw.  $H_1$  ersetzt und anschließend werden die Werte an den Positionen  $L_1$  bzw.  $H_1$  durch die Werte an den

15 Positionen  $L_2$  bzw.  $H_2$  ersetzt. Die Werte in den Listen  $L$  und  $H$  werden also jeweils um eine Position „nach unten“ verschoben, wobei der „unterste“ Wert an der Position  $L_0$  bzw.  $H_0$  herausfällt. Nun wird der Zustandswert  $Z = 5$  an den Positionen  $L_2$  und  $H_2$  abgespeichert.

Ein zweiter ( $n = 2$ ) Zustandswert  $Z = 7$  wird nun erfasst und entsprechend dem obigen Ver-

20 fahren an den Positionen  $H_2$  und  $L_1$  abgespeichert. Dabei werden wieder die Werte ab diesen Positionen eine Position nach unten verschoben bis wieder die Werte an den Positionen  $L_0$  bzw.  $H_0$  herausfallen.

Ein dritter ( $n = 3$ ) Zustandswert  $Z = -3$  wird nun erfasst und entsprechend dem obigen

25 Verfahren an den Positionen  $H_0$  und  $L_2$  abgespeichert. Dabei werden wieder die Werte ab diesen Positionen eine Position nach unten verschoben. Wieder fallen die Werte an den Positionen  $L_0$  bzw.  $H_0$  heraus.

In dem in Figur 5 dargestellten Ausführungsbeispiel wird nun davon ausgegangen, dass kein weiterer Zustandswert  $Z$  mehr erfasst wird. Analog zu den Schritten 150 und 160 aus Figur 4 und den dort angegebenen, beispielhaften Werten für die Endwerte  $B_i$  wird nun der Wert  $j$  ermittelt. Dieser ergibt sich zu  $j = 1$ . Folglich werden die Werte an den Positionen  $L(2-1) = L1$  und  $H(2-1) = H2$  für die Bildung des Mittelwertes  $M$  verwendet, der sich somit zu  $M = 5$  ergibt.

In Figur 6 ist ein weiteres Beispiel für einen Ablauf des Einsortierens von erfassten Zustandswerten  $Z$  in die Listen  $L$  und  $H$  dargestellt. Dabei geschieht das Einsortieren wieder analog zu der Beschreibung der Schritte 120 und 130 des in Figur 4 dargestellten, erfindungsgemäßen Verfahrens.

In diesem Beispiel werden jedoch mehr als  $(2k+2)$  Zustandswerte  $Z$  erfasst und in die Listen  $L$  und  $H$  einsortiert. Zunächst werden, wie bereits in Figur 5 beschrieben, die ersten beiden Zustandswerte  $Z = 5$  und  $Z = 7$  in die Listen  $L$  und  $H$  einsortiert. Anschließend werden weitere  $(2k+m-1)$  Zustandswerte  $Z$  für ein  $m > 2$  einsortiert, was in Figur 6 jedoch lediglich durch den punktierten Übergang von  $(n = 2)$  nach  $(n = 2k + m)$  dargestellt ist.

Der  $(2k+m)$ . erfasste Zustandswert  $Z = 7$  wird nun analog zu den Schritten 120 und 130 aus Figur 4 behandelt und gegebenenfalls in die Listen  $H$  und  $L$  einsortiert. Dabei ergibt sich für die Liste  $L$ , dass alle dort abgespeicherten Werte kleiner sind, als  $Z = 7$ . Deshalb wird der Zustandswert  $Z = 7$  nicht in der Liste  $L$  abgespeichert. Jedoch ergibt eine Prüfung der in der Liste  $H$  abgespeicherten Werte, dass der Wert  $Z = 7$  an der Position  $H1$  abgespeichert werden soll, nachdem der Wert an der Position  $H0$  durch den Wert an der Position  $H1$  ersetzt wurde. Damit fällt wieder der Wert an der Position  $H0$  aus der Liste heraus.

Analog dazu wird ein weiterer, letzter Zustandswert  $Z = -3$  erfasst und an die Position  $L0$  abgespeichert. Da nun mehr als  $(2k+2)$  Zustandswerte erfasst wurden, werden die an den

Positionen L0 und H0 abgespeicherten Werte zur Bildung des Mittelwertes M verwendet. Folglich ist eine fehlertolerante Synchronisation der Uhr 15 mit der Fehlertoleranz  $k = 2$  möglich.

- 5 Bei den in den Figuren 4 und 5 dargestellten Beispielen und ebenso bei dem in Figur 4 dargestellten Ausführungsbeispiel ist es denkbar, das Einsortieren beispielsweise dahingehend zu ändern, dass für einen erfassten Zustandswert Z zunächst geprüft wird, ob er kleiner bzw. größer als der an der Position L0 bzw. H0 abgespeicherte Wert ist und erst wenn dies der Fall ist damit zu beginnen, eine geeignete Position für das Abspeichern zu finden. Damit kann
- 10 eine Ausführungszeit des erfindungsgemäßen Verfahren verringert werden, insbesondere dann, wenn die Anzahl der erfassten Zustandswerte n häufig deutlich größer ist als  $(2k+2)$ .

PATENTANSPRÜCHE

1. Verfahren zum zeitlichen Synchronisieren von Uhren (15), die über ein Kommunikationsmedium (5) kommunizierenden Knoten (10) zugeordnet sind, gekennzeichnet durch die nachfolgenden Schritte:

- 5 - zumindest für die zu synchronisierenden Knoten (10): Erfassen (110) von Zustandswerten, welche abhängig sind von einer Zeitbasis der Knoten (10);
- für alle erfassten Zustandswerte: Einsortieren (120) des erfassten Zustandswertes an eine entsprechende Position in einer ersten, (k+1) Positionen umfassenden Liste (L), falls der erfasste Zustandswert kleiner als das (k+1)-kleinste Element oder kleiner gleich dem
- 10 (k+1)-kleinsten Element der Liste (L) ist und wobei k eine vorgebbare Fehlertoleranz ist;
- für alle erfassten Zustandswerte: Einsortieren (130) des erfassten Zustandswertes an eine entsprechende Position in einer zweiten, (k+1) Positionen umfassenden Liste (H), falls der erfasste Zustandswert größer als das (k+1)-größte Element oder größer gleich dem
- (k+1)-größten Element der Liste (H) ist;
- 15 - Bilden (160) eines Mittelwertes (M) aus dem (k+1) kleinsten Element der ersten Liste (L) und dem (k+1) größten Element der zweiten Liste (H), falls  $n \geq (2k+2)$  gilt, wobei n die Anzahl der erfassten Zustandswerte ist;
- Ermitteln (170) eines Korrekturwertes (K) in Abhängigkeit des Mittelwertes (M); und
- Korrigieren (180) der zu synchronisierenden Uhren (15) dahingehend, dass ein aktueller
- 20 Zustandswert dieser Uhr (15) den Korrekturwert berücksichtigt.

2. Verfahren nach Anspruch 1,

dadurch gekennzeichnet,

- 5 dass das Einsortieren (120, 130) der ermittelten Zustandswerte in der ersten Liste (L) und/oder in der zweiten Liste (H) sequentiell durchgeführt wird.

3. Verfahren nach Anspruch 1 oder 2,

dadurch gekennzeichnet,

- 10 dass die erste Liste (L) durch entsprechende Register (L0, L1, ..., Lk) und/oder die zweite Liste (H) durch entsprechende Register (H0, H1, ..., Hk) realisiert wird.

4. Verfahren nach einem der vorangehenden Ansprüche,

dadurch gekennzeichnet,

- 15 dass
- die erste Liste (L) mit Werten initialisiert wird, die größer als der größte zu erwartende Zustandswert sind; und/oder
  - die zweite Liste (H) mit Werten initialisiert wird, die kleiner als der kleinste zu erwartende Zustandswert sind.

20

25

5. Verfahren nach einem der vorangehenden Ansprüche,

5 dadurch gekennzeichnet,

dass

- bei dem Einsortieren (120) eines erfassten Zustandswertes in die erste Liste (L) eine Sortierung bezüglich der Größe der abgespeicherten Zustandswerte erhalten bleibt, so dass stets  $\text{Wert}(L_0) \geq \text{Wert}(L_1) \geq \dots \geq \text{Wert}(L_k)$  gilt, wobei  $L_0, L_1, \dots, L_k$  die (k+1) Positionen der Liste (L) bezeichnen und  $\text{Wert}(L_i)$  der Wert an einer Position ( $L_i$ ) ist; und
- 10 - bei dem Einsortieren (130) eines erfassten Zustandswertes in die zweite Liste (H) eine Sortierung bezüglich der Größe der abgespeicherten Zustandswerte erhalten bleibt, so dass stets  $\text{Wert}(H_0) \leq \text{Wert}(H_1) \leq \dots \leq \text{Wert}(H_k)$  gilt, wobei  $H_0, H_1, \dots, H_k$  die (k+1) Positionen der Liste (H) bezeichnen und  $\text{Wert}(H_i)$  der Wert an einer Position ( $H_i$ ) ist.

15

6. Verfahren nach einem der vorangehenden Ansprüche,

dadurch gekennzeichnet,

dass ein Zustandswert (Z) in Abhängigkeit der folgenden Schritte an einer Position ( $L_i$ ) der ersten Liste (L) abgespeichert wird:

- 20 - die Positionen ( $L_0, L_1, \dots, L_k$ ) werden nach einer Position ( $L_i$ ) der ersten Liste (L) durchsucht, so dass gilt:
  - $\text{Wert}(L_0) \geq \text{Wert}(L_1) \geq \dots \geq \text{Wert}(L_i) \geq Z \geq \text{Wert}(L_{i+1}) \geq \dots \geq \text{Wert}(L_k)$ ;
  - wird keine solche Position ( $L_i$ ) gefunden, so wird der Zustandswert (Z) verworfen;
  - wird eine solche Position ( $L_i$ ) gefunden, so wird für alle Positionen  $\{L_j \mid 0 \leq j < i\}$ , der an der Position ( $L_j$ ) abgespeicherte Wert  $\text{Wert}(L_j)$  durch den an der Position  $L_{j+1}$  abgespeicherten Wert  $\text{Wert}(L_{j+1})$  ersetzt und der Zustandswert (Z) an der Position ( $L_i$ ) der Liste (L) abgespeichert.

25



7. Verfahren nach einem der vorangehenden Ansprüche,

dadurch gekennzeichnet,

5 dass ein Zustandswert (Z) in Abhängigkeit der folgenden Schritte an einer Position (H<sub>i</sub>) der zweiten Liste (H) abgespeichert wird:

- die Positionen (H<sub>0</sub>, H<sub>1</sub>, ..., H<sub>k</sub>) werden nach einer Position (H<sub>i</sub>) der zweiten Liste (H) durchsucht, so dass gilt:

- $\text{Wert}(H_0) \leq \text{Wert}(L_1) \leq \dots \leq \text{Wert}(L_i) \leq Z \leq \text{Wert}(L_{i+1}) \leq \dots \leq \text{Wert}(L_k)$ ;

10 - wird keine solche Position (H<sub>i</sub>) gefunden, so wird der Zustandswert (Z) verworfen;

- Wird eine solche Position (H<sub>i</sub>) gefunden, so wird für alle Positionen  $\{H_j \mid 0 \leq j < i\}$ , der an der Position H<sub>j</sub> abgespeicherte Wert  $\text{Wert}(H_j)$  durch den an der Position H<sub>(j+1)</sub> abgespeicherten Wert  $\text{Wert}(H_{j+1})$  ersetzt und der Zustandswert (Z) an der Position (H<sub>i</sub>) der Liste (H) abgespeichert.

15

8. Verfahren nach einem der vorangehenden Ansprüche,

dadurch gekennzeichnet,

dass die folgenden Schritte durchgeführt werden:

- in Abhängigkeit einer Fehlertoleranz (k) wird eine Menge (B) von vorgebbaren Endwerten

20  $\{B_0, B_1, \dots, B_{(k-1)}\}$  derart vorgegeben, dass

- $B_0 = 0$ ;  $B_i \leq B_{i+1}$ , für alle  $i \in \{0, 1, \dots, (k-1)\}$ ; und

- $2j < B(j)$ , für alle  $j \in \{1, \dots, (k)\}$  gilt;

- falls  $B_k > n$  gilt, wird in Abhängigkeit der Anzahl n der ermittelten Zustandswerte ein Wert i für  $i \in \{0, 1, \dots, (k-1)\}$  gewählt, so dass die Bedingung  $B_i \leq n < B_{i+1}$  gilt;

25 - falls  $B_k \leq n$  gilt, wird  $i = k$  gewählt; und

- der Mittelwert (M) wird aus den an den Positionen L<sub>(k-i)</sub> und H<sub>(k-i)</sub> abgespeicherten Werten  $\text{Wert}(L_{(k-j)})$  und  $\text{Wert}(H_{(k-j)})$  gebildet.

9. Verfahren nach einem der vorangehenden Ansprüche,  
dadurch gekennzeichnet,
- 5 dass die folgenden Werte vorgegeben werden:
- Fehlertoleranz  $k = 2$ ;
  - Endwert  $B1 = 3$ ; und
  - Endwert  $B2 = 8$ .
- 10 10. Knoten (10), der mittels eines Kommunikationsmediums (5) mit anderen Knoten (10) kommuniziert,  
dadurch gekennzeichnet,  
dass der Knoten (10)
- eine Uhr (15) aufweist;
- 15 - Mittel zum Erfassen von Zustandswerten aufweist, wobei die Zustandswerte von einer Zeitbasis des Knotens (10) und/oder von einer Zeitbasis der anderen Knoten abhängen;
- eine  $(k+1)$  Positionen umfassende erste Liste (L) und eine  $(k+1)$  Positionen umfassende zweite Liste (H) aufweist;
  - Mittel zum Einsortieren (120) eines erfassten Zustandswertes an eine entsprechende
- 20 Position der ersten Liste (L) aufweist;
- Mittel zum Einsortieren (130) eines erfassten Zustandswertes an eine entsprechende Position der zweiten Liste (H) aufweist;
  - Mittel zum Bilden (160) eines Mittelwertes (M) aus einem Element der ersten Liste (L) und einem Element der zweiten Liste (H) aufweist;
- 25 - Mittel zum Bilden eines Korrekturwertes (K) aufweist; und
- Mittel zum Korrigieren der Uhr (15) aufweist.

11. Knoten (10) nach Anspruch 10,

dadurch gekennzeichnet,

dass in dem Knoten (10) ein Verfahren nach einem der Ansprüche 1 bis 9 abläuft.

5

12. Kommunikationssystem (1), das mehrere über ein Kommunikationsmedium (5)

kommunizierende Knoten (10) aufweist,

dadurch gekennzeichnet,

dass mindestens ein Knoten (10)

- 10 - eine Uhr (15) aufweist;
- Mittel zum Erfassen von Zustandswerten aufweist;
  - eine (k+1) Positionen umfassende erste Liste (L) und eine (k+1) Positionen umfassende zweite Liste (H) aufweist;
  - Mittel zum Einsortieren (120) eines erfassten Zustandswertes an eine entsprechende
- 15 Position der ersten Liste (L) aufweist;
- Mittel zum Einsortieren (130) eines erfassten Zustandswertes an eine entsprechende Position der zweiten Liste (H) aufweist;
  - Mittel zum Bilden (160) eines Mittelwertes (M) aus einem Element der ersten Liste (L) und einem Element der zweiten Liste (H) aufweist;
- 20 - Mittel zum Bilden eines Korrekturwertes (K) aufweist; und
- Mittel zum Korrigieren der Uhr (15) aufweist.

13. Kommunikationssystem (1) nach Anspruch 12,

dadurch gekennzeichnet,

- 25 dass in mindestens einem Knoten (10) ein Verfahren nach einem der Ansprüche 1 bis 9 abläuft.

- 
14. Computerprogramm, das auf einem Recheng Gerät, insbesondere auf einem  
5 Mikroprozessor, ablauffähig ist,  
dadurch gekennzeichnet,  
dass das Computerprogramm zur Durchführung eines Verfahrens nach einem der Ansprüche 1  
bis 9 programmiert ist, wenn es auf dem Recheng Gerät abläuft.
- 10 15. Computerprogramm nach Anspruch 10,  
dadurch gekennzeichnet,  
dass das Computerprogramm auf einem Speicherelement, insbesondere auf einem Random-  
Access-Memory (RAM), einem Read-Only-Memory (ROM) oder einem Flash-Memory,  
abgespeichert ist.
- 15

ZUSAMMENFASSUNG

## Verfahren zum zeitlichen Synchronisieren von Uhren

Um in einem Kommunikationssystem (1) eine zeitliche Synchronisation von Uhren besonders schnell und effizient durchzuführen, wird ein Verfahren vorgeschlagen, das die nachfolgenden

5 Schritte aufweist:

- erfassen von Zustandswerten, welche abhängig sind von einer Zeitbasis der Knoten (10);
- einsortieren jedes erfassten Zustandswertes an eine Position in einer ersten, (k+1) Positionen umfassenden Liste L, falls der erfasste Zustandswert kleiner als oder gleich dem (k+1)-kleinsten Element der Liste L ist, wobei keine vorgebbare Fehlertoleranz ist;
- 10 - einsortieren des erfassten Zustandswertes an eine Position in einer zweiten, (k+1) Positionen umfassenden Liste H, falls der erfasste Zustandswert größer oder gleich dem (k+1)-größten Element der Liste H ist;
- bilden eines Mittelwertes aus dem (k+1)-kleinsten Element der Liste L und dem (k+1)-größten Element der Liste H, falls die Anzahl der erfassten Zustandswerte größer oder
- 15 gleich  $(2k+2)$  ist;
- ermitteln eines Korrekturwertes in Abhängigkeit des Mittelwertes; und
- korrigieren eines aktuellen Zustandswertes der zu synchronisierenden Uhren.

(Fig. 1)

Fig. 1

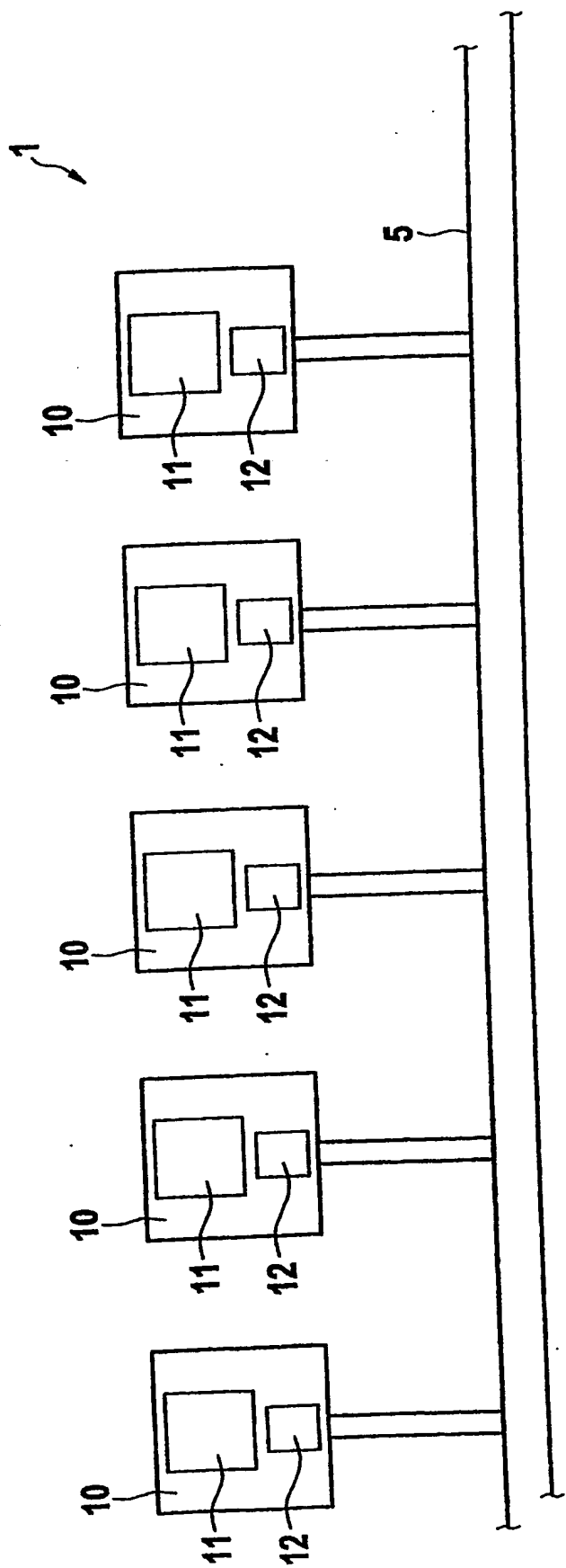


Fig. 2a

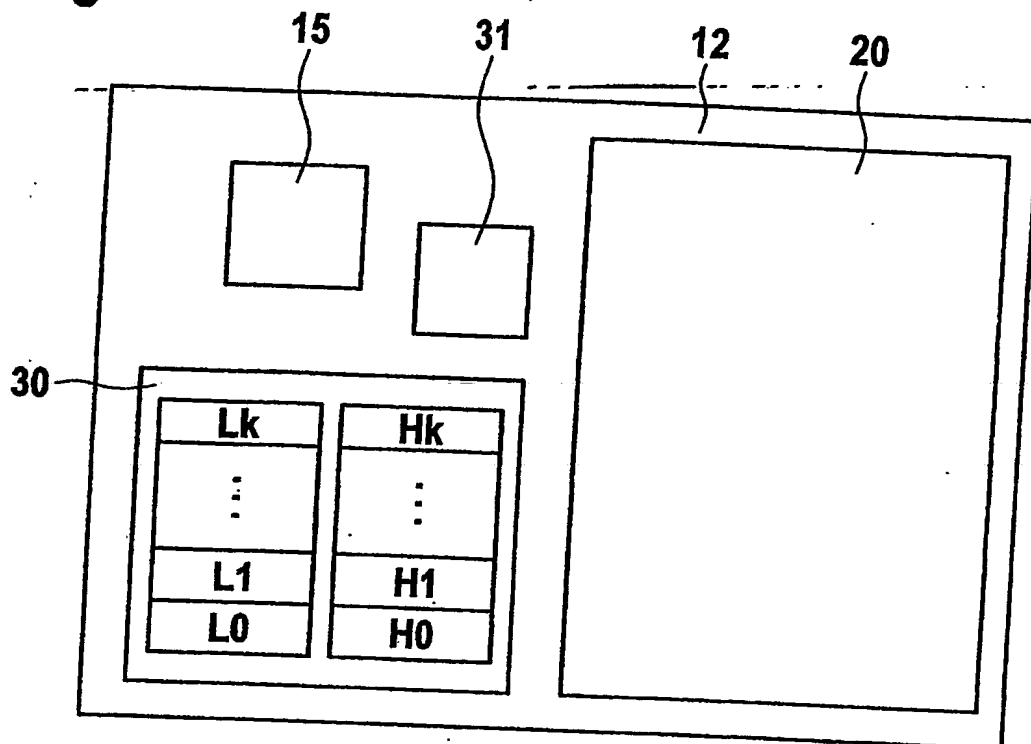


Fig. 2b

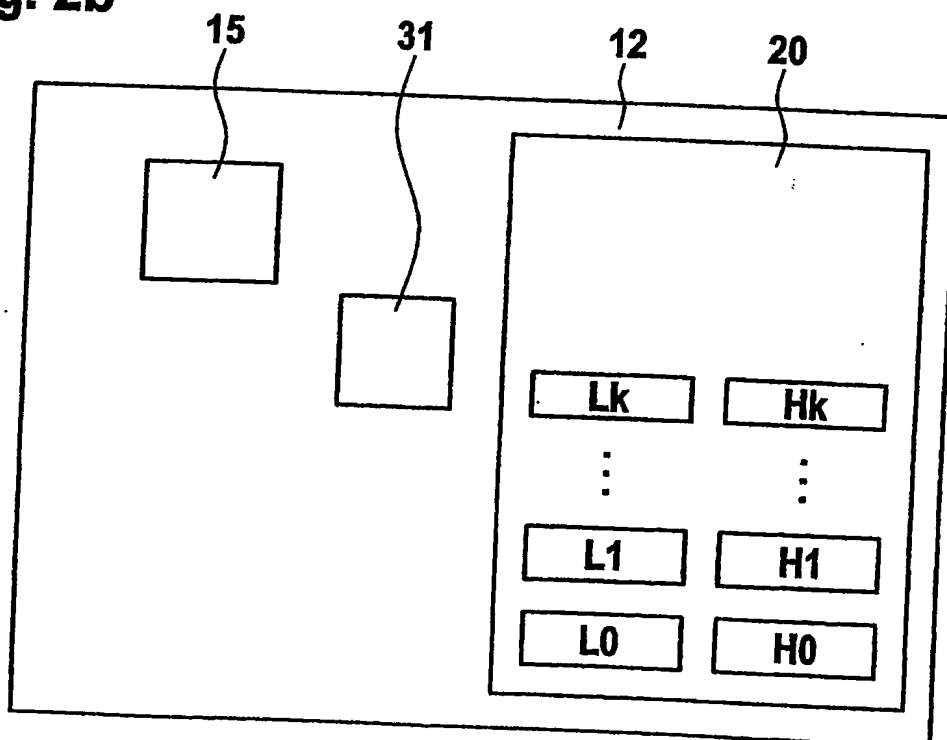
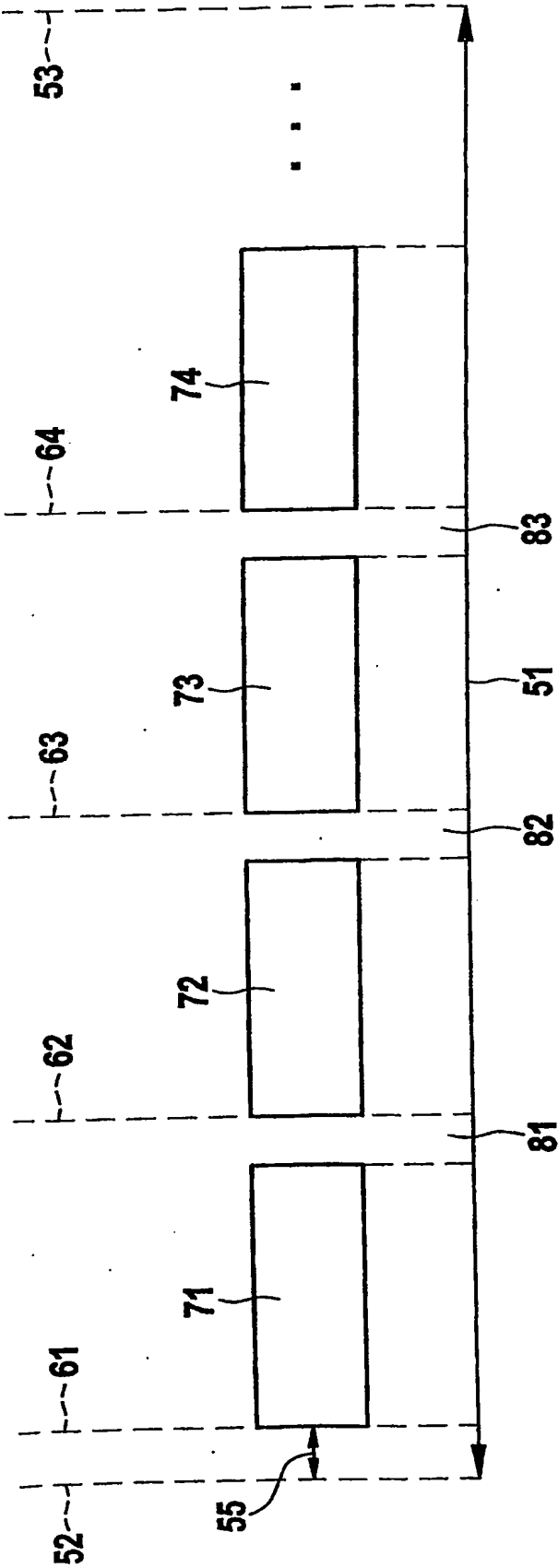


Fig. 3





4 / 6

Fig. 4

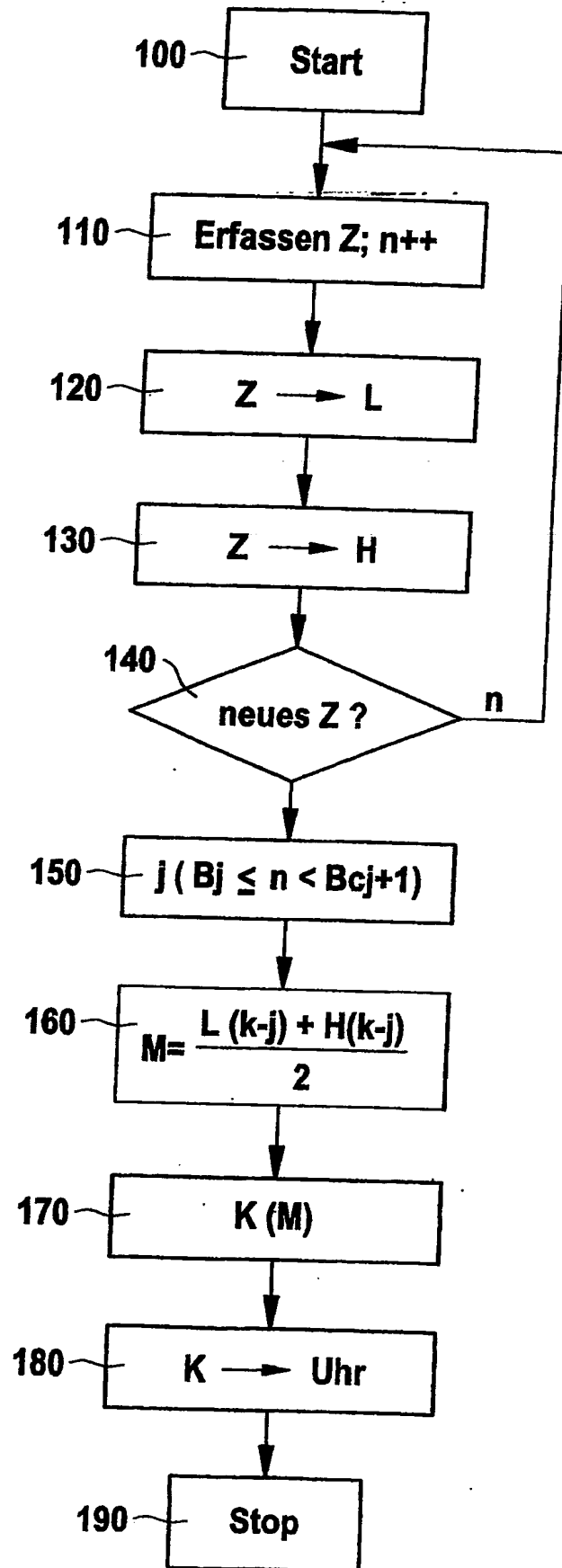


Fig. 5

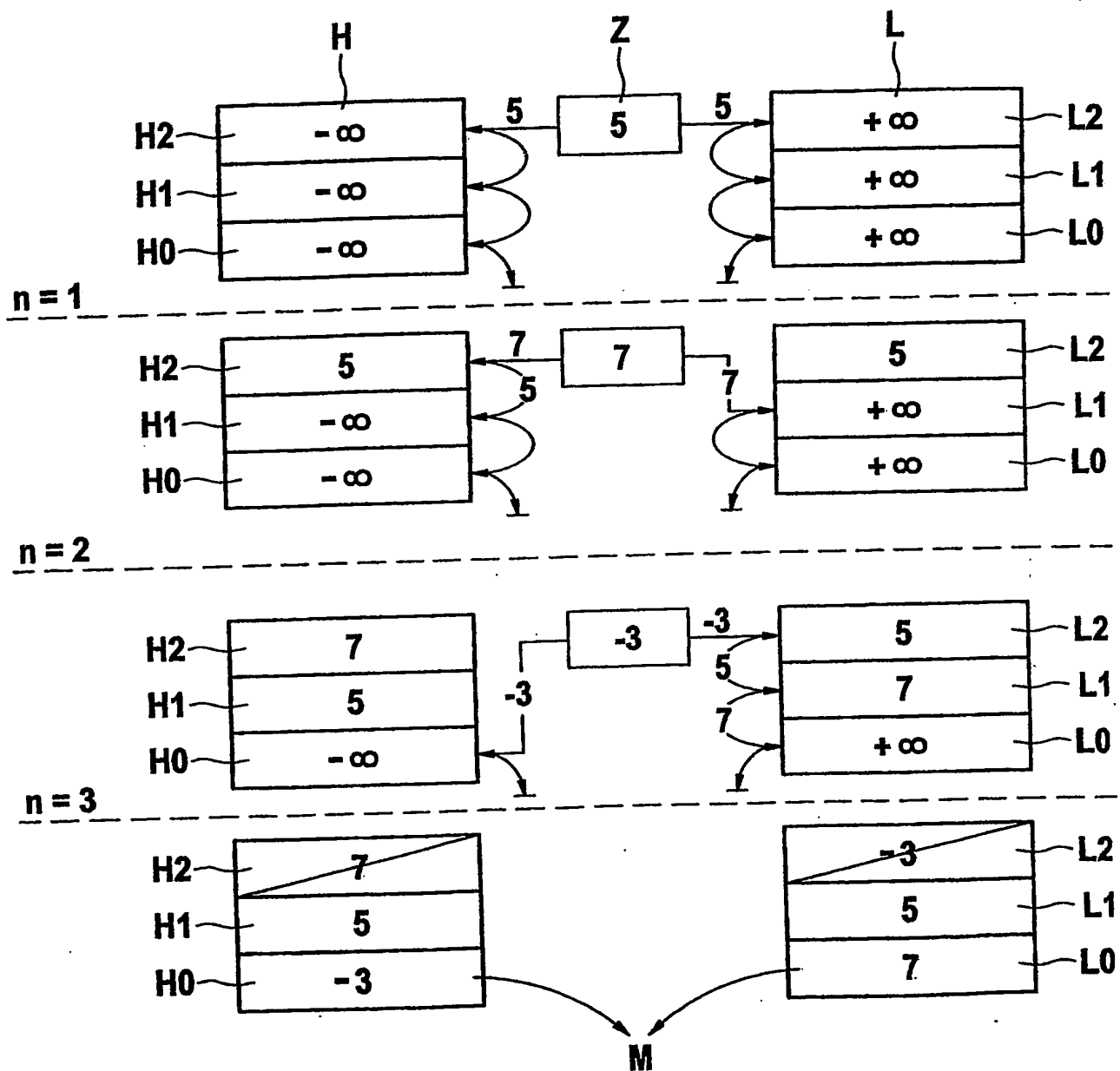
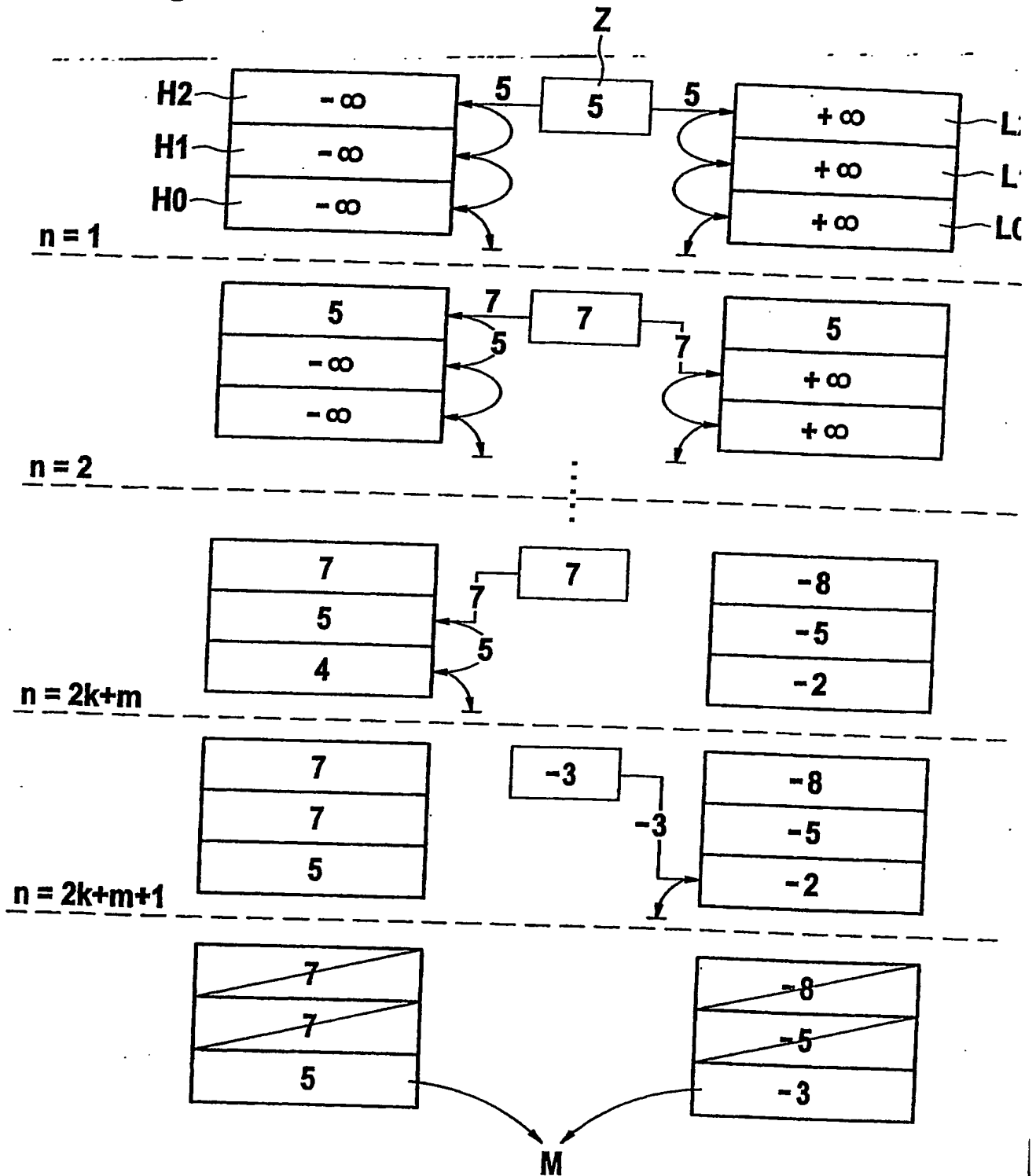


Fig. 6



PCT/IB2004/050511

